

DOCKET NO.: 51876P557

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re the Application of:

SANG-HOON HONG, ET AL.

Application No.:

Filed:

For: **Semiconductor Memory Device for
Enhancing Refresh Operation in
High Speed Data Access**

Art Group:

Examiner:

Commissioner for Patents
P.O, Box 1450
Alexandria, VA 22313-1450

REQUEST FOR PRIORITY

Sir:

Applicant respectfully requests a convention priority for the above-captioned application, namely:

COUNTRY	APPLICATION NUMBER	DATE OF FILING
Republic of Korea	2003-27654	30 April 2003

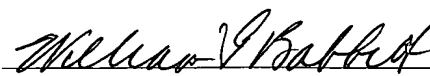
☒ A certified copy of the document is being submitted herewith.

Respectfully submitted,

Blakely, Sokoloff, Taylor & Zafman LLP

Dated: 12/30/03

12400 Wilshire Boulevard, 7th Floor
Los Angeles, CA 90025
Telephone: (310) 207-3800



William Thomas Babbitt, Reg. No. 39,591



별첨 사본은 아래 출원의 원본과 동일함을 증명함.

This is to certify that the following application annexed hereto is a true copy from the records of the Korean Intellectual Property Office.

출원번호 : 10-2003-0027654
Application Number

출원년월일 : 2003년 04월 30일
Date of Application APR 30, 2003

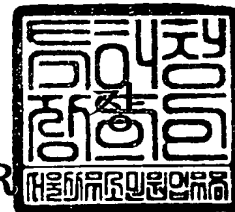
출원인 : 주식회사 하이닉스반도체
Applicant(s) Hynix Semiconductor Inc.



2003 년 10 월 24 일

특 허 청

COMMISSIONER



【서지사항】

【서류명】	특허출원서
【권리구분】	특허
【수신처】	특허청장
【참조번호】	0006
【제출일자】	2003.04.30
【발명의 명칭】	리프레쉬 동작이 향상된 고속 데이터 액세스를 위한 반도체 메모리 장치
【발명의 영문명칭】	SEMICONDUCTOR MEMORY DEVICE FOR HIGH SPEED DATA ACCESS WITH ENHANCEMENT OF REFRESH OPERATION
【출원인】	
【명칭】	주식회사 하이닉스반도체
【출원인코드】	1-1998-004569-8
【대리인】	
【명칭】	특허법인 신성
【대리인코드】	9-2000-100004-8
【지정된변리사】	변리사 정지원, 변리사 원석희, 변리사 박해천
【포괄위임등록번호】	2000-049307-2
【발명자】	
【성명의 국문표기】	홍상훈
【성명의 영문표기】	HONG, Sang Hoon
【주민등록번호】	700930-1064113
【우편번호】	467-860
【주소】	경기도 이천시 부발읍 신하리 청구 아파트 101-1302
【국적】	KR
【발명자】	
【성명의 국문표기】	안진홍
【성명의 영문표기】	AHN, Jin Hong
【주민등록번호】	581124-1110419
【우편번호】	431-070
【주소】	경기도 안양시 동안구 평촌동 130-1 영풍아파트 101-1408
【국적】	KR

【발명자】

【성명의 국문표기】 고재범
【성명의 영문표기】 K0,Jae Bum
【주민등록번호】 760926-1026025
【우편번호】 467-862
【주소】 경기도 이천시 부발읍 신하6리 삼익세라믹아파트 102-208
【국적】 KR

【발명자】

【성명의 국문표기】 김세준
【성명의 영문표기】 KIM,Se Jun
【주민등록번호】 740209-1069419
【우편번호】 463-500
【주소】 경기도 성남시 분당구 구미동 까치마을 선경아파트 107-1002
【국적】 KR

【심사청구】

청구

【취지】

특허법 제42조의 규정에 의한 출원, 특허법 제60조의 규정에 의한 출원심사를 청구합니다. 대리인
 특허법인 신성 (인)

【수수료】

【기본출원료】	20 면	29,000 원
【가산출원료】	10 면	10,000 원
【우선권주장료】	0 건	0 원
【심사청구료】	6 항	301,000 원
【합계】		340,000 원

【첨부서류】

1. 요약서·명세서(도면)_1통

【요약서】

【요약】

본 발명은 태그블럭을 구비하여 고속으로 데이터의 액세스가 가능한 메모리 장치에서 리프레쉬 동작시간을 최소화하여 안정적인 고속동작이 가능한 메모리 장치를 제공하기 위한 것으로, 이를 위해 본 발명은 각각 M개의 워드라인을 가지며, 입력되는 로우어드레스에 대응하도록 구비된 N개의 단위셀블럭에, 추가적으로 M개의 워드라인을 가지기 위해 추가의 단위셀블럭을 더 포함하여 N+1개의 단위셀블럭으로 구성되는 셀블럭; 상기 (N+1) 개의 워드라인 중에서, 적어도 어느 한 워드라인을 예비워드라인으로 할지의 여부에 대한 정보를 저장하기 위한 예비셀블럭 테이블; 상기 로우어드레스를 입력받아 N개의 단위셀블럭을 선택하기 위한 논리적 셀블럭어드레스를 감지하고, 이를 상기 N+1개의 단위셀블럭중 하나를 선택하기 위한 물리적 셀블럭어드레스로 변환하여 출력하며, 데이터 액세스를 위해 상기 물리적 셀블럭어드레스가 변환될 때에 이를 누적하여 저장하는 태그블럭; 및 상기 물리적 셀블럭어드레스에 의해 선택된 단위셀블럭에서의 한 워드라인과, 상기 워드라인에 대한 -상기 예비셀블럭 테이블로부터 제공되는 정보에 의해 정해지는- 예비워드라인을 활성화시키기 위해 상기 태그블럭과 상기 예비셀블럭 테이블을 제어하는 제어수단을 구비하며, 상기 태그블럭에 누적되어 저장되는 물리적 셀블럭어드레스에 대응하는 워드라인에 한하여 리프레쉬 동작을 수행하는 메모리 장치를 제공한다.

【대표도】

도 5

1020030027654

출력 일자: 2003/10/29

【색인어】

반도체, 메모리, 리프레쉬, 예비워드라인, 태그메모리.

【명세서】**【발명의 명칭】**

리프레쉬 동작이 향상된 고속 데이터 액세스를 위한 반도체 메모리 장치{SEMICONDUCTOR MEMORY DEVICE FOR HIGH SPEED DATA ACCESS WITH ENHANCEMENT OF REFRESH OPERATION}

【도면의 간단한 설명】

도1은 종래기술에 의한 메모리 장치의 블럭구성도.

도2는 도1에 도시된 태그블럭의 블럭구성도.

도3은 도1에 도시된 예비셀블럭 테이블의 블럭구성도.

도4는 도1에 도시된 메모리 장치의 동작을 나타내는 동작 흐름도.

도5는 본 발명의 바람직한 실시예에 따른 메모리 장치의 블럭구성도.

도6은 도5에 도시된 태그블럭을 나타내는 블럭구성도.

도7은 도5에 도시된 메모리 장치의 리프레쉬 동작을 나타내는 동작 흐름도.

【발명의 상세한 설명】**【발명의 목적】****【발명이 속하는 기술분야 및 그 분야의 종래기술】**

<8> 본 발명은 반도체 메모리 장치에 관한 것으로, 특히 태그메모리를 사용하여 고속으로 데이터 액세스가 가능한 반도체 장치의 리프레쉬 동작에 관한 것이다.

<9> 근래에 중앙처리장치(CPU)의 동작속도는 메모리 장치(DRAM)의 동작속도를 능가할 정도로 현저히 향상되어 왔으며, 그 결과 메모리 장치의 동작속도가 중앙처리장치의 동작속도보다 상대적으로 느려 여러가지 문제점이 발생하고 있다. 이러한 문제점을 극복하기 위해 보다 고속으로 데이터를 입출력하기 위한 다양한 구조의 메모리 장치가 개발되고 있다.

<10> 도1은 본 출원인이 2003년 2월 21일에 출원한 고속으로 데이터 액세스가 가능한 메모리 장치(출원번호 2003-11121)의 블록구성도이다.

<11> 도1을 참조하여 살펴보면, 고속으로 데이터 액세스가 가능한 메모리 장치는 각각 256개의 워드라인을 가지며, 입력되는 로우어드레스에 대응하도록 구비된 8개의 단위셀블럭에, 추가적으로 256개의 워드라인을 가지기 위해 추가의 단위셀블럭을 더 포함하여 9개의 단위셀블럭으로 구성되는 셀블럭(500)과, 셀블럭에(500) 구비되는 $(8+1) \times 256$ 개의 워드라인 중에서, 적어도 어느 한 워드라인을 예비워드라인으로 할지의 여부에 대한 정보를 저장하기 위한 예비셀블럭 테이블(410)과, 상기 로우어드레스를 입력받아 8개의 단위셀블럭을 선택하기 위한 논리적 셀블럭어드레스를 감지하고, 이를 9개의 단위셀블럭중 하나를 선택하기 위한 물리적 셀블럭어드레스로 변환하여 출력하기 위한 태그블럭(430)과, 물리적 셀블럭어드레스에 의해 선택된 단위셀블럭에서의 한 워드라인과, 상기 워드라인에 대한 -상기 예비셀블럭 테이블로부터 제공되는 정보에 의해 정해지는- 예비워드라인을 활성화시키기 위해 태그블럭(430)과 예비셀블럭 테이블(410)을 제어하는 제어부(420)를 구비한다.

<12> 여기서 제어부(420)는 9개의 단위셀블럭중에서 선택된 하나의 단위셀블럭에 연속적으로 제1 및 제2 데이터가 액세스될 때, 제1 데이터에 대한 재저장동작은 제1 데이터를 액세스하기 위해 활성화된 워드라인에 대응하여 지정된 예비워드라인을 통해 이루어지도록 제어하게 된다.

- <13> 또한 도에 도시된 메모리 장치의 제어부(420)에 입력되는 어드레스는 8 × 256개의 워드라인에 대응하고, 추가의 단위셀블럭의 구비되는 워드라인의 수만큼인 256개의 워드라인을 예비 워드라인으로 지정하게 된다. 예비워드라인은 한 단위셀블럭에 있는 256개의 워드라인으로 고정적으로 정해지는 것은 아니고 메모리 장치가 동작중에서 계속해서 위치가 변하게 되는데, 이에 대해서는 뒤에서 자세히 설명한다.
- <14> 도2는 도1에 도시된 태그블럭의 블럭구성도이다.
- <15> 도2를 참조하여 살펴보면, 태그블럭(430)은 9개의 단위셀블럭(510 ~ 590)에 각각 구비된 256개의 워드라인이 어떤 논리적 셀블럭에 대응되는 것인지를 각각 저장하고 있는 9개의 단위 태그테이블(432a~432i)과, 입력되는 논리적 셀블럭어드레스(Cur_LBA)와, 입력되는 로컬어드레스(Cur_RA)에 대응되어 9개의 단위태그테이블(432a~432i)에서 출력되는 정보 -로컬어드레스(Cur_RA)에 의해 선택된 워드라인이 어떤 논리적셀블럭에 관한 것에 관한 9개의 데이터 정보-를 각각 비교하기 위한 9개의 비교부(433a~433i)와, 9개의 비교부(433a~433i)에서 비교한 정보를 인코딩하여 물리적 셀블럭어드레스(Cur_PBA)를 출력하기 위한 인코더(434)와 9개의 단위태그테이블(432a~432i)과, 9개의 비교부(433a~433i)와 인코더(434)를 제어하기 위한 태그제어부(436)를 구비한다.
- <16> 태그블럭(430)은 예비셀블럭 테이블(420)에서 출력되는 정보(Extra_BA) -로컬어드레스(Cur_RA)에 대응하는 워드라인의 예비워드라인이 9개의 단위셀블럭(500)중에서 어떤 단위셀블럭에 있는지에 대한 정보-를 디코딩하여, 9개의 단위태그테이블(432a~432i)중에서 하나의 단위태그테이블을 선택하기 위한 신호를 출력하는 셀블럭어드레스 디코더부(431)를 더 구비한다. 또한 태그블럭(430)은 현재 동작클럭(CK)의 물리적 셀블럭어드레스(Cur_PBA)를 한클럭 지연시

켜 이전 클럭의 물리적 셀블럭어드레스(Pre_PBA)를 출력하기 위한 태그지연부(435)을 구비한다

<17> 여기서 각각의 단위태그테이블(432a~432i)은 256개의 레지스터를 구비하고, 논리적 셀블럭어드레스가 8개이므로 각각의 레지스터는 3비트로 구성된다. 예를 들어 제1 단위태그테이블(432a)은 제1 단위셀블럭에 구비된 256개의 워드라인이 어떤 논리적 셀블럭에 대한 것인지를 저장하고 있고, 제2 단위태그테이블(432b)는 제2 단위셀블럭에 구비된 256개의 워드라인이 각각 어떤 논리적 셀블럭에 대한 것인지를 저장하고 있는 것이다.

<18> 각각의 단위태그테이블(432a~432i)에서 첫번째 레지스터(0)는 제1 단위셀블럭의 워드라인 'WL0'의 논리적 셀블럭어드레스를 저장하고 두번째 레지스터(1)는 제1 단위셀블럭의 워드라인 'WL1'의 논리적 셀블럭어드레스를 저장하고, 제256번째 레지스터(255)는 제1 단위셀블럭의 워드라인 'WL255'의 논리적 셀블럭어드레스를 저장한다.

<19> 예컨대 도2에 도시된 제1 단위태그테이블(432a)을 참조하여 살펴보면, 첫번째 레지스터에 저장된 데이터 '1'으로 인하여 제1 단위셀블럭의 워드라인(WL0)은 제2 논리적 단위셀블럭의 워드라인(WL0)에 대응하는 데이터라는 뜻이고, 255번째 레지스터에 저장된 데이터 '7'로 인하여, 제1 단위셀블럭의 워드라인(WL255)은 제8 논리적 단위셀블럭의 워드라인(WL255)에 대응하는 데이터인 것이다.

<20> 도3은 도1에 도시된 예비셀블럭 테이블(410)의 블럭구성도이다.

<21> 도3을 참조하여 살펴보면, 예비셀블럭 테이블(410)은 256개의 예비워드라인이 9개의 물리적 단위셀블럭중에서 어떤 단위셀블럭에 있는지에 대한 정보를 저장하기 위한 256개의 레지스터(411)를 구비한다. 256개의 워드라인(WL)에 대한 예비워드라인 정보를 저장하기 위해 예비

셀블럭 테이블(410)에는 256개의 레지스터(411)가 있고, 각각의 레지스터(411)는 9개의 단위셀블럭에 대한 물리적 셀블럭어드레스에 대한 정보를 저장해야 하기 때문에 각각 4비트로 구현된다.

<22> 예비셀블럭 테이블(410)에 저장된 내용을 살펴보면, 워드라인(WL0)의 예비워드라인은 제3 단위셀블럭의 워드라인 'WL0'이라는 것이고(첫번째 레지스터(0) 참조), 워드라인(WL3)의 예비워드라인은 제5 단위셀블럭의 워드라인 'WL3'이라는 뜻이다(네번째 레지스터(3) 참조). 메모리 장치의 동작중에 레지스터(411)에 저장된 내용은 계속 업데이트되며, 업데이트 될 때마다 256개의 예비워드라인에 대한 정보는 계속 바뀌게 된다.

<23> 또한, 예비셀블럭 테이블(410)는 현재 실행단계의 로컬어드레스(Cur_RA)를 입력받아 현재 실행되고 있는 로컬어드레스(Cur_RA)에 대한 예비워드라인이 어떤 단위셀블럭에 지정되었는지를 알려주는 예비블럭 신호(Extra_BA)를 태그블럭(430)으로 출력한다. 또한, 예비셀블럭 테이블(410)는 이전 실행단계의 로컬어드레스(Pre_RA)와, 이전 실행단계의 물리적 셀블럭어드레스(Pre_PBA)를 입력받아 예비블럭 업데이터 신호(EBT_UPDATE)에 응답하여 레지스터(411)를 업데이터하게 된다.

<24> 도4는 도1에 도시된 메모리 장치의 동작을 나타내는 동작 흐름도이다. 이하 도1 내지 도4를 참조하여 도4에 도시된 메모리 장치의 동작에 대해서 설명한다.

<25> 도1에 도시된 메모리 장치는 입력되는 어드레스에 대응하는 단위셀블럭이 8개인 경우에 하나의 단위셀블럭을 더 구비하여 총 8+1개의 단위셀블럭을 구비한다. 추가적으로 구비되는 하나의 단위셀블럭은 동일한 단위셀블럭에 연속해서 데이터 액세스가 일어날 때 이전 명령어에 의해 액세스된 데이터의 재저장을 위한 것이다.



- <26> 도4에서는 8+1개의 단위셀블럭중에서 제1 단위셀블럭에 연속적으로 데이터가 액세스되고, 입력되는 어드레스에 의해 워드라인 WL0, WL1이 연속해서 활성화되는 것으로 가정하였다.
- <27> 먼저 제1 명령어(CD0)에 의해 8+1개의 단위셀블럭 중에서 선택된 제1 단위셀블럭의 제1 워드라인(WL0)이 활성화된다. 이어서, 제1 워드라인(WL0)에 대응되는 K개의 데이터를 제1 단위셀블럭에 구비된 센스앰프로 감지 증폭한다. 이어서 제1 워드라인(WL0)에 대응하는 K개의 데이터를 데이터래치부(450)로 이동시켜 래치시킨다. 이어서, 제1 워드라인(WL0)에 대응하여 감지 증폭된 K개의 데이터가 래치된 제1 단위셀블럭의 센스앰프를 강제프리차지시킨다.
- <28> 이어서 제1 명령어(CD0)가 리드명령어인 경우에는 데이터래치부(20)에 래치된 K개의 데이터중 선택된 하나의 데이터를 제1 명령어(CD0)에 대응하는 출력데이터로 출력시키고, 제1 명령어(CD))가 라이트명령어인 경우에는 제1 명령어(CD))에 대응하여 입력되는 데이터로 데이터래치부(20)에 래치된 K개의 데이터중 선택된 하나의 데이터를 덮어쓰게 된다.
- <29> 이어서 제1 워드라인(WL0)에 대응하는 예비워드라인으로 지정된 제3 단위셀블럭의 워드라인(WL0)을 활성화시킨다. 이어서 데이터래치부(450)에 래치된 K개의 데이터를 제3 단위셀블럭의 워드라인(WL0)에 대응되는 K개의 단위셀에 재저장한다.
- <30> 이어서 제2 명령어(CD1)에 대응하여 선택된 제1 단위셀블럭의 제2 워드라인(WL1)을 활성화시킨다. 이어서 제2 워드라인(WL1)에 대응하는 K개의 데이터를 감지증폭한다.
- <31> 여기서 제1 명령어(CD0)에 대응하여 감지, 증폭된 K개의 데이터가 재저장 동작과 제2 명령어(CD1)에 대응하여 K개의 데이터가 감지, 증폭되는 동작은 실질적으로 같은 타이밍에 이루어진다.

- <32> 이렇게 데이터를 액세스 함으로서 이전명령어에 의해 감지, 증폭된 K개의 데이터가 재저장되는 타이밍에 상관없이 다음 명령어에 의해 데이터를 감지, 증폭할 수 있게 되어, 데이터 재저장시간만큼 데이터엑세스 시간을 줄일 수 있게 되는 것이다.
- <33> 한편 입력되는 명령어에 의해 활성화되는 워드라인에 대해서는 하나의 예비워드라인이 필요하게 되는데, 이는 예비셀블럭 테이블에 저장되어 있다. 예비워드라인으로 지정되는 워드라인의 수는 하나의 단위셀블럭에 구비되는 워드라인의 수인 256개이다. 하나에 단위셀블럭 구비된 모든 워드라인이 예비워드라인이 고정되는 것이 아니고, 메모리 장치의 동작중에 계속 바뀌게 되는 것이다. 이는 어드레스에 대응하는 단위셀블럭보다 추가적으로 하나의 단위셀블럭을 구비함으로서 가능하게 된다.
- <34> 이어서 제1 데이터 및 제2 데이터가 제1 및 제2 단위셀블럭에서 교대로 액세스되는 인터리빙모드로 동작하는 것에 대해서 살펴본다.
- <35> 먼저 제1 데이터를 8+1개의 단위셀블럭중에서 선택된 제1 단위셀블럭에서 감지 증폭한 다음 데이터래치부(450)에 래치시킨다. 이어서 제1 단위셀블럭에 제1 데이터를 재저장시키고, 상기 재저장 타이밍과 실질적으로 동일한 타이밍에 상기 8+1개의 단위셀블럭중에서 선택된 제2 단위셀블럭에서 상기 제2 데이터를 감지 증폭하여 데이터래치부(450)에 래치시킨다.
- <36> 따라서 도1에 메모리장치는 데이터가 하나의 셀에 계속해서 액세스되는 경우에는 이전 데이터에 대한 재저장은 지정된 예비워드라인에 대응하는 단위셀블럭에서 수행하고, 데이터가 단위셀에 교대로 액세스되는 경우에는 인터리빙모드로 동작함으로서, 액세스되는 패턴에 상관없이 고속으로 데이터를 액세스 할 수 있게 되는 것이다.

- <37> 도1에 도시된 메모리 장치는 입력되는 셀블럭어드레스에 대응하는 8개의 단위셀블럭에다 추가적으로 하나의 단위셀블럭을 더 구비하기 때문에, 입력되는 셀블럭어드레스는 논리적 셀블럭어드레스를 뜻하는 것이고, 이를 실제 9개의 물리적 단위셀블럭 중 하나를 선택하는 물리적셀어드레스로 변환하는 변환부가 필요하게 되는데, 제어부(420)에서 태그블럭(430)을 제어하여 셀블럭어드레스 변환 동작을 하게 된다.
- <38> 또한, 물리적셀블럭어드레스를 저장하고 있는 저장부가 필요하게 되는데, 이 때의 저장부가 태그블럭이며, 전술한 바와 같이 256개의 예비워드라인에 대한 정보를 저장하고 있는 저장부는 예비셀블럭테이블이다. 8+1개의 단위셀블럭이 있고, 하나의 단위셀블럭에는 256개의 워드라인이 구비된다면, 태그블럭에는 9개의 단위태그테이블이 구비되고 하나의 단위태그테이블에는 256 ×8비트의 메모리셀이 필요하며, 예비블럭테이블에는 256 ×4비트의 메모리셀이 필요하다.
- <39> 한편, 도1에 도시된 메모리 장치는 단위 저장매체로 캐패시터를 사용하고 있기 때문에 리프레쉬 동작이 필수적이다.
- <40> 도1에 도시된 바와 같이, 태그블럭을 사용하여 고속으로 데이터 액세스가 가능한 메모리 장치의 리프레쉬 동작은 이전의 다른 통상적인 메모리 장치에서와 마찬가지로 데이터가 액세스되지 않는 구간동안 각 단위셀블럭별로 리프레쉬 동작을 수행하게 된다.
- <41> 즉, 데이터가 액세스되지 않는 구간동안 각 단위셀블럭의 모든 워드라인을 인을 순차적으로 활성화시키고, 활성화된 워드라인에 대응하는 데이터를 감지 증폭한다. 감지, 증폭된 데이터는 다시 활성화된 워드라인에 대응하는 단위셀에 재저장된다.

<42> 따라서 모든 단위셀블럭의 워드라인을 활성화시키고, 그에 대응하는 단위셀에 저장된 데이터를 리프레쉬하게되면 많은 시간이 소요된다. 실제로 현재 워드라인별로는 한번도 활성화되지 않아서 의미있는 데이터가 저장되어 있지 않는 경우도 있을 수 있다.

<43> 실제로 데이터가 저장되어 있던지 관계없이 모든 워드라인을 활성화시켜 리프레쉬 동작을 수행하는 것은 많은 파워를 소모하게 되고, 또한 메모리 장치의 고속 동작에 크게 방해가 되는 문제점을 가지고 있다.

【발명이 이루고자 하는 기술적 과제】

<44> 본 발명은 태그블럭을 구비하여 고속으로 데이터의 액세스가 가능한 메모리 장치에서 리프레쉬 동작시간을 최소화하여 안정적인 고속동작이 가능한 메모리 장치를 제공하는 것을 특징으로 한다.

【발명의 구성 및 작용】

<45> 본 발명은 상기의 과제를 해결하기 위해 각각 M개의 워드라인을 가지며, 입력되는 로우 어드레스에 대응하도록 구비된 N개의 단위셀블럭에, 추가적으로 M개의 워드라인을 가지기 위해 추가의 단위셀블럭을 더 포함하여 N+1개의 단위셀블럭으로 구성되는 셀블럭; 상기 (N+1) 개의 워드라인 중에서, 적어도 어느 한 워드라인을 예비워드라인으로 할지의 여부에 대한 정보를 저장하기 위한 예비셀블럭 테이블; 상기 로우어드레스를 입력받아 N개의 단위셀블럭을 선택하기 위한 논리적 셀블럭어드레스를 감지하고, 이를 상기 N+1개의 단위셀블럭중 하나를 선택하기 위한 물리적 셀블럭어드레스로 변환하여 출력하며, 데이터 액세스를 위해 상기 물리적 셀블럭

어드레스가 변환될 때에 이를 누적하여 저장하는 태그블럭; 및 상기 물리적 셀블럭어드레스에 의해 선택된 단위셀블럭에서의 한 워드라인과, 상기 워드라인에 대한 -상기 예비셀블럭 테이블로부터 제공되는 정보에 의해 정해지는- 예비워드라인을 활성화시키기 위해 상기 태그블럭과 상기 예비셀블럭 테이블을 제어하는 제어수단을 구비하며, 상기 태그블럭에 누적되어 저장되는 물리적 셀블럭어드레스에 대응하는 워드라인에 한하여 리프레쉬 동작을 수행하는 메모리 장치를 제공한다.

<46> 또한, 본 발명은 상기의 메모리 장치를 리프레쉬하는 방법에 있어서, 리프레쉬 모드에 진입하는 제1 단계; 상기 태그블럭에 구비된 $M \times (N+1)$ 개의 리프레쉬용 제2 레지스터를 판독하여 상기 $N+1$ 개의 단위셀블럭에 구비되는 $M \times (N+1)$ 개의 워드라인 중에서 데이터가 저장되어 있는 워드라인을 감지하는 제2 단계; 및 상기 제2 단계에서 감지된 워드라인에 대응하는 단위셀에 대해서만 리프레쉬 동작을 수행하는 제3 단계를 포함하는 메모리 장치의 리프레쉬 방법을 제공한다.

<47> 이하, 본 발명이 속하는 기술분야에서 통상의 지식을 가진 자가 본 발명의 기술적 사상을 용이하게 실시 할 수 있을 정도로 상세히 설명하기 위하여, 본 발명의 가장 바람직한 실시예를 첨부된 도면을 참조하여 설명하기로 한다.

<48> 도5는 본 발명의 바람직한 실시예에 따른 메모리 장치의 블럭구성도이다.

<49> 도5를 참조하여 살펴보면, 본 실시예에 따른 메모리 장치는 각각 256개의 워드라인을 가지며, 입력되는 로우어드레스에 대응하도록 구비된 8개의 단위셀블럭(510 ~ 580)에, 추가적으로 256개의 워드라인을 가지기 위해 추가의 단위셀블럭(590)을 더 포함하여 8+1 개의 단위셀블

력으로 구성되는 셀블럭(500)과, $(8+1) \times 256$ 개의 워드라인 중에서, 적어도 어느 한 워드라인을 예비워드라인으로 할지의 여부에 대한 정보를 저장하기 위한 예비셀블럭 테이블(450)과, 로우 어드레스를 입력받아 8개의 단위셀블럭중 하나를 선택하기 위한 논리적 셀블럭어드레스를 감지하고, 이를 $8+1$ 개의 단위셀블럭(500)중 하나를 선택하기 위한 물리적 셀블럭어드레스로 변환하여 출력하며, 데이터 액세스를 위해 상기 물리적 셀블럭어드레스가 변환될 때에 이를 누적하여 저장하는 태그블럭(460)과, 물리적 셀블럭어드레스에 의해 선택된 단위셀블럭에서의 한 워드라인과, 워드라인에 대한 - 예비셀블럭 테이블(470)로부터 제공되는 정보에 의해 정해지는- 예비워드라인을 활성화시키기 위해 태그블럭(460)과 예비셀블럭 테이블(470)을 제어하는 제어부(420)를 구비하며, 태그블럭(460)에 누적되어 저장되는 물리적 셀블럭어드레스에 대응하는 워드라인에 한하여 리프레쉬 동작을 수행한다.

<50> 예비셀블럭 테이블(470)은 256개의 예비워드라인이 상기 $8+1$ 개의 물리적 단위셀블럭(510 ~ 590) 중에서 어떤 단위셀블럭에 있는지에 대한 정보를 저장하기 위해서, 각각 3+1비트의 데이터를 저장할 수 있는 256개의 레지스터(471)를 구비한다.

<51> 도6은 도5에 도시된 태그블럭을 나타내는 블럭구성도이다.

<52> 도6을 참조하여 살펴보면, 태그블럭(460)은 $8+1$ 개의 단위셀블럭(510 ~ 590)에 각각 구비된 256개의 워드라인이 어떤 논리적셀블럭에 대응되는 지를 각각 저장하기 위해 각각 256개의 레지스터를 구비하는 $8+1$ 개의 단위태그테이블(432a ~ 432i)과, 입력되는 논리적셀블럭어드레스(Cur_LBA)와, 입력되는 로컬어드레스(Cur_RA)에 대응되어 상기 9개의 단위태그테이블(432a~432i)에서 출력되는 정보 -상기 로컬어드레스(Cur_RA)에 의해 선택된 워드라인이 어떤 논리적셀블럭에 관한 것에 관한 9개의 데이터 정보- 와 각각 비교하기 위한 9개의 비교부(433a~433i)와, 9개의 비교부(433a~433i)에서 비교한 정보를 인코딩하여 물리적 셀블럭어드레

스(Cur_PBA)를 출력하기 위한 인코더(434)와, 9개의 단위태그테이블(432a~432i)과, 9개의 비교부(433a~433i)와 인코더(434)를 제어하기 위한 태그제어부(436)를 구비한다.

- <53> 단위태그테이블(예컨대 423a)에 구비되는 256개의 레지스터는 각각 8개의 단위셀블럭(432a~432i)에 대응하는 논리적 셀블럭어드레스를 저장하기 위한 3비트의 제1 레지스터(X)와, 제1 레지스터(X)의 데이터가 업데이트되는 것을 감지하기 위한 리프레쉬용 제2 레지스터(Y)를 구비한다. 여기서 제1 레지스터(X)가 3비트인 것은 입력되는 로우어드레스에 대응하는 논리적 셀블럭이 8개이기 때문이다.
- <54> 도7은 도5에 도시된 메모리 장치의 리프레쉬 동작을 나타내는 동작 흐름도이다. 이하에서는 도5 내지 도7을 참조하여 본 실시예에 따른 메모리 장치의 리프레쉬 동작을 살펴본다.
- <55> 본 실시예에 따라 태그메모리를 구비하여 고속으로 동작하는 메모리 장치를 리프레쉬하는 방법은 먼저 리프레쉬 모드에 진입한 다음에(S1), 태그블럭(460)에 구비된 $256 \times (8+1)$ 개의 리프레쉬용 제2 레지스터를 판독하여 $8+1$ 개의 단위셀블럭(510 ~ 590)에 구비되는 $256 \times (8+1)$ 개의 워드라인 중에서 데이터가 저장되어 있는 워드라인을 감지하고(S2), 예비셀블럭 테이블(470)에 구비된 256개의 레지스터를 감지하여, 256개의 예비워드라인이 $8+1$ 개의 단위셀블럭(510 ~ 590)의 어디에 있는지 판독하고(S3), 리프레쉬 동작을 수행할 때에는 데이터가 저장된 워드라인에 한하여 리프레쉬 동작을 수행하되, 예비워드라인으로 지정된 워드라인에 대해서는 리프레쉬 동작을 스킵(skip)하도록 한다(S4).
- <56> 종래의 태그 메모리를 구비한 고속 메모리 장치에서 리프레쉬 동작을 할 때에는 리드명령어나 라이트명령어에 따른 데이터 액세스가 일어나지 않는 리프레쉬 구간에서 데이터가 저장되어 있는지 여부에 상관없이 모든 단위셀블럭에 대해서 리프레쉬 동작을 수행하였기 때문에 리프레쉬 동작에 따른 많은 시간이 소요되었다.



- <57> 그러나 본 발명에 의해서 각 단위셀블럭의 워드라인별로 데이터가 저장되었는지를 감지하고, 데이터가 저장되어 있는 워드라인에 한하여 리프레쉬 동작을 수행하기 때문에 리프레쉬 동작에 소요되는 시간을 크게 줄일 수 있게 되었다.
- <58> 이를 위하여 본 발명에서는 하나의 워드라인에 데이터가 저장되어 있는 지를 판단하기 위한 리프레쉬용 제2 레지스터(Y)를 태그메모리에 추가로 구비하는 것이다. 태그블럭(460)에 구비되는 단위태그테이블(432a ~ 432i)에는 각각 3비트를 가지는 256개의 제1 레지스터(X)와, 추가적으로 한비트를 더 저장할 수 있는 리프레쉬용 제2 레지스터(Y)를 구비하게 되는 것이다.
- <59> 제1 레지스터(X)가 3비트인 것은 8개의 논리적 셀블럭에 대한 정보를 저장하기 위한 것이다. 만약 논리적 셀블럭이 N개 라면 이에 대한 제1 레지스터는 논리적 셀블럭 어드레스를 저장하기 위해서는 아래의 수학식1과 같이 $\log_2 N$ 보다 큰 X비트수를 가지면 된다.
- <60> 【수학식 1】 $X \geq \log_2 N$
- <61> 따라서 본 실시예에 따른 메모리 장치의 단위태그테이블(432a)에는 각각 3비트의 데이터를 저장할 수 있는 245개의 제1 레지스터(X)와, 1비트의 리프레쉬용 제2 레지스터를 구비하여, 총 4 × 256개의 데이터를 저장할 수 있는 메모리셀을 구비하게 된다.
- <62> 전술한 바와 같이, 태그 메모리를 구비하는 고속메모리 장치에서는 입력되는 로우어드레스를 논리적 셀블럭어드레스와 하나의 단위셀블럭에 구비된 256개의 워드라인중 하나를 선택하기 위한 로컬어드레스로 분리하고, 논리적 셀블럭어드레스는 태그블럭(460)에 의해서 물리적 셀블럭어드레스로 변환된다. 논리적 셀블럭어드레스를 물리적 셀블럭어드레스로 변환할 때에 단위태그테이블의 제1 레지스터 하나가 선택이 되는데(예를 들어 432a의 첫번째 레지스터(0)),

선택된 제1 레지스터에 데이터를 업데이트 할 때에 그에 대응하는 제2 레지스터에 '1'을 저장한다.

<63> 데이터 액세스가 한번 수행될 때마다 제1 레지스터 하나가 업데이트 되며, 업데이트 될 때마다 그에 대응하는 제2 레지스터에는 '1'이 저장된다.

<64> 이 후 리프레쉬 모드에서 리프레쉬 동작을 수행할 때에 '1'이 저장된 제2 레지스터에 대응하는 워드라인이 데이터가 의미 있는 저장되어 있는 워드라인이므로, '1'이 저장된 제2 레지스터에 대응하는 워드라인에 대해서만 리프레쉬를 수행하면 되는 것이다.

<65> 한편, 전술한 바와 같이 태그메모리를 구비하는 고속 메모리 장치는 예비워드라인이라는 개념을 사용하기 때문에 9개의 단위셀블럭에 구비되어 있는 256 개의 워드라인중에서 256개는 데이터 재저장을 위한 예비워드라인으로 사용되고, 이에 관한 정보는 예비태그테이블(470)에 저장되어 있다.

<66> 예비워드라인으로 지정된 워드라인에 대응하는 단위셀에는 의미있는 데이터가 저장되어 있지 않기 때문에, 예비워드라인에 대해서는 리프레쉬 동작을 수행할 필요가 없게 된다.

<67> 따라서 9개의 단위태그테이블에 구비된 256 개의 제2 레지스터중에서 '1'이 저장된 제2 레지스터에 대응하는 워드라인에 대해서만 리프레쉬 동작을 수행하되, 예비태그테이블에서 지정한 예비워드라인에 대해서는 리프레쉬 동작을 스킵하여도 되는 것이다.

<68> 결론적으로 본 발명에 의해서 태그메모리를 구비하여, 데이터 재저장을 지정된 예비워드라인으로 이동시켜 함으로서 고속동작이 가능한 메모리 장치의 리프레쉬 동작시간을 크게 줄일 수 있어 보다 더우 고속으로 메모리 장치를 동작시킬 수 있다. 또한 리프레쉬동작시간이 줄어들음으로서 소모되는 파워도 줄일 수 있게 되었다.



<69> 이상에서 설명한 본 발명은 전술한 실시예 및 첨부된 도면에 의해 한정되는 것이 아니고, 본 발명의 기술적 사상을 벗어나지 않는 범위 내에서 여러 가지 치환, 변형 및 변경이 가능하다는 것이 본 발명이 속하는 기술분야에서 통상의 지식을 가진 자에게 있어 명백할 것이다.

【발명의 효과】

<70> 본 발명에 의해서 본 발명에 의해서 태그메모리를 구비하는 고속 메모리 장치에서 리프레쉬 동작시간을 크게 줄일 수 있어 보다 더 빠르게 메모리 장치를 고속을 동작시킬 수 있다.

【특허청구범위】

【청구항 1】

각각 M개의 워드라인을 가지며, 입력되는 로우어드레스에 대응하도록 구비된 N개의 단위 셀블럭에, 추가적으로 M개의 워드라인을 가지기 위해 추가의 단위셀블럭을 더 포함하여 N+1개의 단위셀블럭으로 구성되는 셀블럭;

상기 (N+1) 개의 워드라인 중에서, 적어도 어느 한 워드라인을 예비워드라인으로 할지의 여부에 대한 정보를 저장하기 위한 예비셀블럭 테이블;

상기 로우어드레스를 입력받아 N개의 단위셀블럭을 선택하기 위한 논리적 셀블럭어드레스를 감지하고, 이를 상기 N+1개의 단위셀블럭중 하나를 선택하기 위한 물리적 셀블럭어드레스로 변환하여 출력하며, 데이터 액세스를 위해 상기 물리적 셀블럭어드레스가 변환될 때에 이를 누적하여 저장하는 태그블럭; 및

상기 물리적 셀블럭어드레스에 의해 선택된 단위셀블럭에서의 한 워드라인과, 상기 워드라인에 대한 -상기 예비셀블럭 테이블로부터 제공되는 정보에 의해 정해지는- 예비워드라인을 활성화시키기 위해 상기 태그블럭과 상기 예비셀블럭 테이블을 제어하는 제어수단

을 구비하며, 상기 태그블럭에 누적되어 저장되는 물리적 셀블럭어드레스에 대응하는 워드라인에 한하여 리프레쉬 동작을 수행하는 메모리 장치.

【청구항 2】

제 1 항에 있어서,

상기 태그블럭은

N+1 개의 단위셀블럭에 각각 구비된 M개의 워드라인이 어떤 논리적셀블럭에 대응되는 지를 각각 저장하기 위해 각각 M개의 레지스터를 구비하는 N+1개의 단위태그테이블;

상기 로우어드레스에서 감지된 논리적 셀블럭어드레스와, 로컬어드레스(단위셀블럭의 한 워드라인을 선택하기 위한 어드레스)에 대응되어 상기 N+1개의 단위태그테이블에서 출력되는 정보 -상기 로컬어드레스에 의해 선택된 워드라인이 어떤 논리적 단위셀블럭에 대응되는 것인지에 관한 N+1개의 데이터 정보- 를 각각 비교하기 위한 N+1개의 비교수단;

상기 N+1개의 비교수단에서 비교한 정보를 인코딩하여 상기 물리적 셀블럭어드레스를 출력하기 위한 인코딩수단; 및

상기 N+1개의 단위태그테이블과 상기 N+1개의 비교수단과 상기 인코딩수단을 제어하기 위한 태그제어부를 구비하는 것을 특징으로 하는 메모리 장치.

【청구항 3】

제 2 항에 있어서,

상기 단위태그테이블에 구비되는 M개의 레지스터는 각각 N개의 단위셀블럭에 대응하는 논리적 셀블럭어드레스를 저장하기 위한 X비트($X \geq \log_2 N$)의 제1 레지스터; 및

상기 제1 레지스터의 데이터가 업데이트되는 것을 감지하기 위한 리프레쉬용 제2 레지스터를 구비하는 것을 특징으로 하는 메모리 장치.

【청구항 4】

제 3 항에 있어서,

상기 예비셀블럭 테이블은

상기 M개의 예비워드라인이 상기 N+1개의 물리적 단위셀블럭중에서 어떤 단위셀블럭에 있는지에 대한 정보를 저장하기 위해서, 각각 X+1 비트의 데이터를 저장할 수 있는 M개의 레지스터를 구비하는 것을 특징으로 하는 메모리 장치.

【청구항 5】

제 4 항의 메모리 장치를 리프레쉬하는 방법에 있어서,

리프레쉬 모드에 진입하는 제1 단계;

상기 태그블럭에 구비된 $M \times (N+1)$ 개의 리프레쉬용 제2 레지스터를 판독하여 상기 N+1개의 단위셀블럭에 구비되는 $M \times (N+1)$ 개의 워드라인 중에서 데이터가 저장되어 있는 워드라인을 감지하는 제2 단계; 및

상기 제2 단계에서 감지된 워드라인에 대응하는 단위셀에 대해서만 리프레쉬 동작을 수행하는 제3 단계

를 포함하는 메모리 장치의 리프레쉬 방법.

【청구항 6】

제 5 항에 있어서,

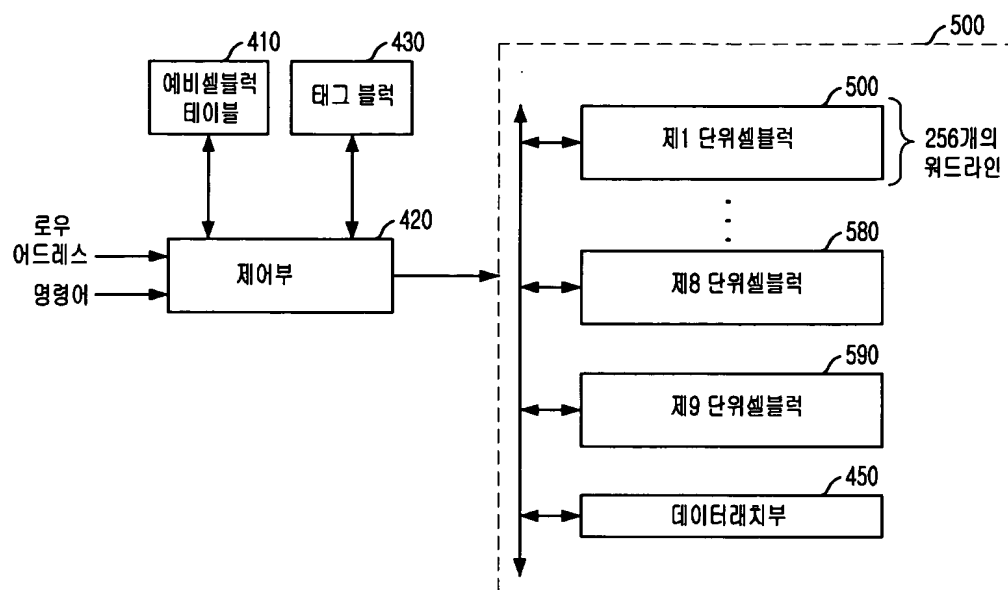
상기 예비셀블럭 테이블에 구비된 M개의 레지스터를 감지하여, M개의 예비워드라인이 상기 N+1개의 단위셀블럭의 어디에 있는지 판독하는 제4 단계를 더 포함하고 상기 제3 단계에서



의 리프레쉬 동작에서 상기 제4 단계에서 판독된 예비워드라인에 대응하는 단위셀에 대해서는 리프레쉬 동작을 스킵하는 것을 특징으로 하는 메모리 장치의 리프레쉬 방법.

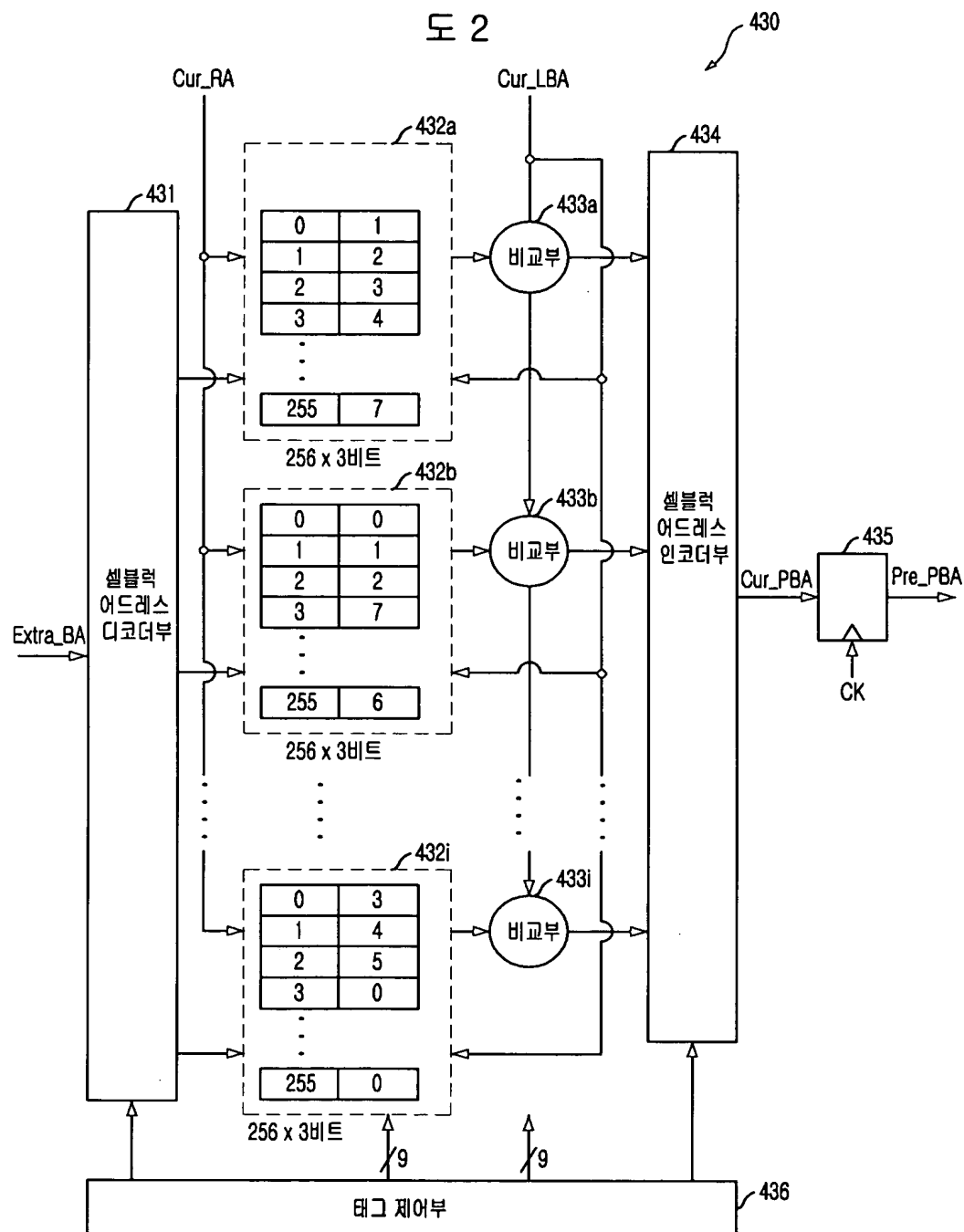
【도면】

【도 1】

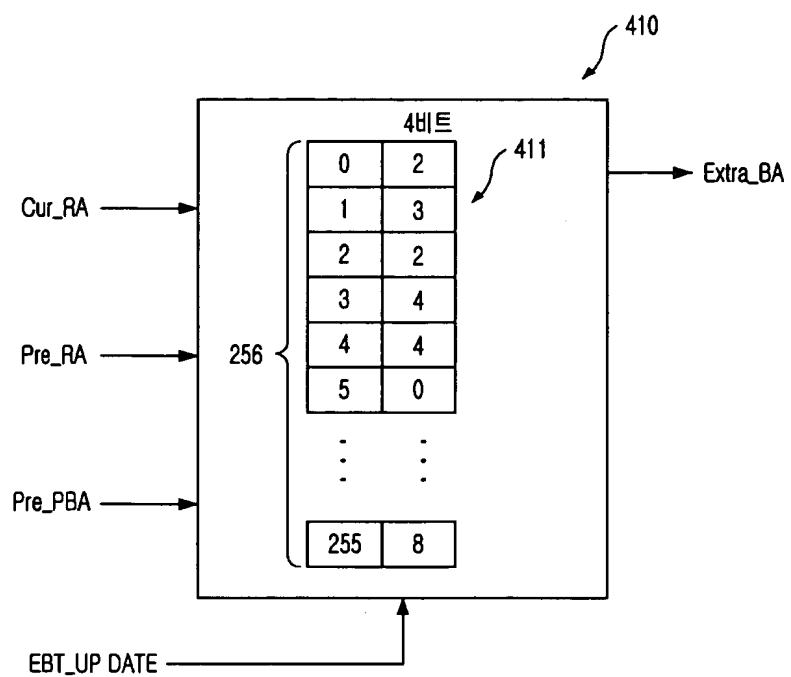




【도 2】

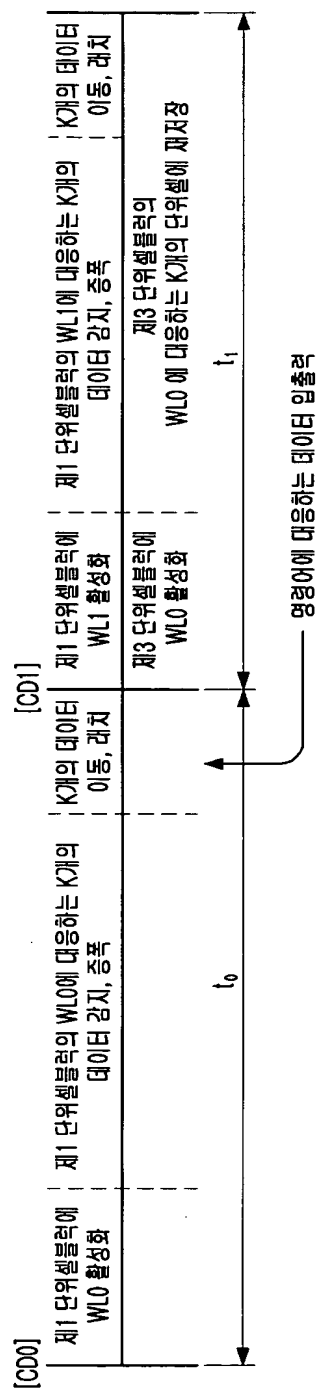


【도 3】

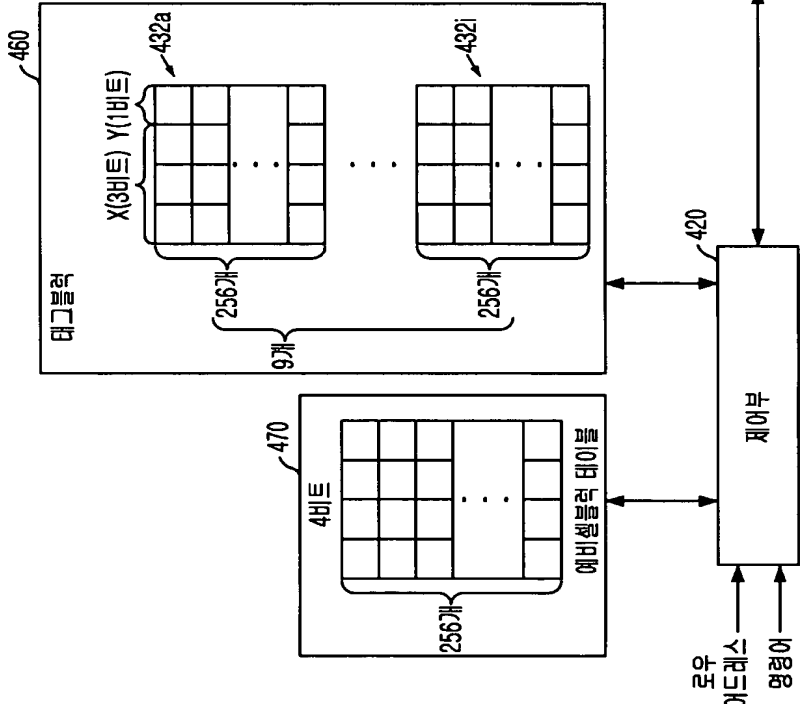




【도 4】

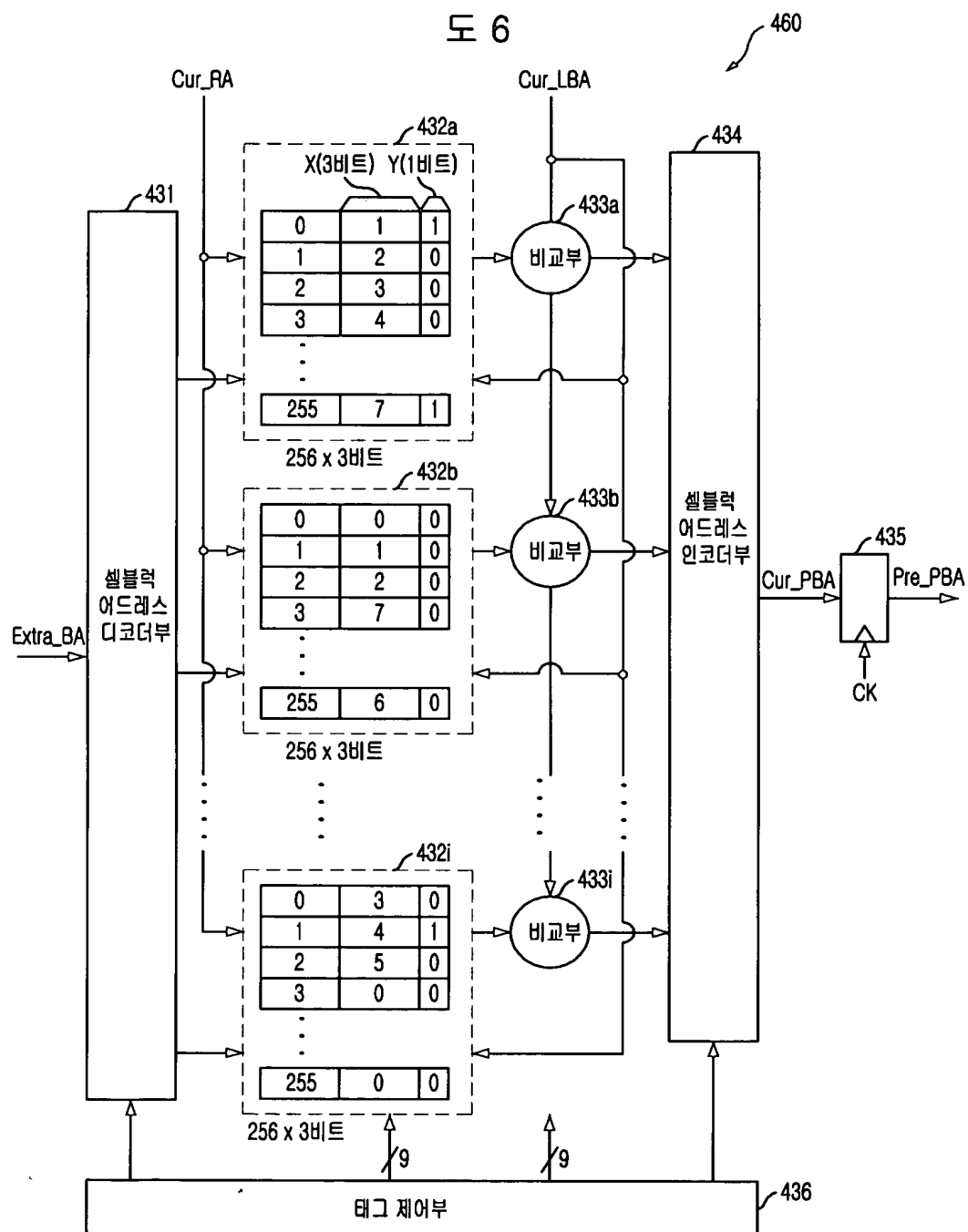


【도 5】





【도 6】



【도 7】

